

Introduction to RBM package

Dongmei Li

April 20, 2026

Clinical and Translational Science Institute, University of Rochester School of Medicine and Dentistry, Rochester, NY 14642-0708

Contents

1 Overview	1
2 Getting started	2
3 RBM_T and RBM_F functions	2
4 Ovarian cancer methylation example using the RBM_T function	6

1 Overview

This document provides an introduction to the RBM package. The RBM package executes the resampling-based empirical Bayes approach using either permutation or bootstrap tests based on moderated t-statistics through the following steps.

- Firstly, the RBM package computes the moderated t-statistics based on the observed data set for each feature using the `lmFit` and `eBayes` function.
- Secondly, the original data are permuted or bootstrapped in a way that matches the null hypothesis to generate permuted or bootstrapped resamples, and the reference distribution is constructed using the resampled moderated t-statistics calculated from permutation or bootstrap resamples.
- Finally, the p-values from permutation or bootstrap tests are calculated based on the proportion of the permuted or bootstrapped moderated t-statistics that are as extreme as, or more extreme than, the observed moderated t-statistics.

Additional detailed information regarding resampling-based empirical Bayes approach can be found elsewhere (Li et al., 2013).

2 Getting started

The RBM package can be installed and loaded through the following R code. Install the RBM package with:

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+   install.packages("BiocManager")
> BiocManager::install("RBM")
```

Load the RBM package with:

```
> library(RBM)
```

3 RBM_T and RBM_F functions

There are two functions in the RBM package: `RBM_T` and `RBM_F`. Both functions require input data in the matrix format with rows denoting features and columns denoting samples. `RBM_T` is used for two-group comparisons such as study designs with a treatment group and a control group. `RBM_F` can be used for more complex study designs such as more than two groups or time-course studies. Both functions need a vector for group notation, i.e., "1" denotes the treatment group and "0" denotes the control group. For the `RBM_F` function, a contrast vector need to be provided by users to perform pairwise comparisons between groups. For example, if the design has three groups (0, 1, 2), the `aContrast` parameter will be a vector such as ("X1-X0", "X2-X1", "X2-X0") to denote all pairwise comparisons. Users just need to add an extra "X" before the group labels to do the contrasts.

- Examples using the `RBM_T` function: `normdata` simulates a standardized gene expression data and `unifdata` simulates a methylation microarray data. The p -values from the `RBM_T` function could be further adjusted using the `p.adjust` function in the `stats` package through the Benjamini-Hochberg method.

```
> library(RBM)
> normdata <- matrix(rnorm(1000*6, 0, 1),1000,6)
> mydesign <- c(0,0,0,1,1,1)
> myresult <- RBM_T(normdata,mydesign,100,0.05)
> summary(myresult)
```

	Length	Class	Mode
<code>ordfit_t</code>	1000	-none-	numeric
<code>ordfit_pvalue</code>	1000	-none-	numeric
<code>ordfit_beta0</code>	1000	-none-	numeric
<code>ordfit_beta1</code>	1000	-none-	numeric
<code>permutation_p</code>	1000	-none-	numeric
<code>bootstrap_p</code>	1000	-none-	numeric

```
> sum(myresult$permutation_p<=0.05)
```

```

[1] 30

> which(myresult$permutation_p<=0.05)

[1] 10 46 67 74 160 217 274 322 399 407 462 500 506 519 523 525 553 585 587
[20] 638 659 695 710 890 943 949 954 977 995 996

> sum(myresult$bootstrap_p<=0.05)

[1] 10

> which(myresult$bootstrap_p<=0.05)

[1] 74 334 352 419 526 656 686 827 890 995

> permutation_adj_p <- p.adjust(myresult$permutation_p, "BH")
> sum(permutation_adj_p<=0.05)

[1] 4

> bootstrap_adj_p <- p.adjust(myresult$bootstrap_p, "BH")
> sum(bootstrap_adj_p<=0.05)

[1] 0

> unifdata <- matrix(runif(1000*7,0.10, 0.95), 1000, 7)
> mydesign2 <- c(0,0,0, 1,1,1,1)
> myresult2 <- RBM_T(unifdata,mydesign2,100,0.05)
> sum(myresult2$permutatioin_p<=0.05)

[1] 0

> sum(myresult2$bootstrap_p<=0.05)

[1] 37

> which(myresult2$bootstrap_p<=0.05)

[1] 2 22 45 59 73 76 89 94 129 146 195 227 247 273 287 303 342 369 379
[20] 453 479 481 494 497 505 508 561 566 585 632 780 795 818 863 897 932 965

> bootstrap2_adj_p <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adj_p<=0.05)

[1] 0

```

- Examples using the RBM_F function: normdata_F simulates a standardized gene expression data and unifdata_F simulates a methylation microarray data. In both examples, we were interested in pairwise comparisons.

```

> normdata_F <- matrix(rnorm(1000*9,0,2), 1000, 9)
> mydesign_F <- c(0, 0, 0, 1, 1, 1, 2, 2, 2)
> aContrast <- c("X1-X0", "X2-X1", "X2-X0")
> myresult_F <- RBM_F(normdata_F, mydesign_F, aContrast, 100, 0.05)
> summary(myresult_F)

              Length Class  Mode
ordfit_t      3000  -none-  numeric
ordfit_pvalue 3000  -none-  numeric
ordfit_beta1  3000  -none-  numeric
permutation_p 3000  -none-  numeric
bootstrap_p   3000  -none-  numeric

> sum(myresult_F$permutation_p[, 1]<=0.05)

[1] 54

> sum(myresult_F$permutation_p[, 2]<=0.05)

[1] 57

> sum(myresult_F$permutation_p[, 3]<=0.05)

[1] 56

> which(myresult_F$permutation_p[, 1]<=0.05)

[1]  6 42 47 53 63 68 82 90 110 207 226 258 260 271 327 333 347 351 361
[20] 374 381 382 395 397 398 422 464 468 477 502 517 576 591 627 631 677 732 733
[39] 741 773 780 781 810 817 820 825 834 871 879 888 922 946 980 989

> which(myresult_F$permutation_p[, 2]<=0.05)

[1]  6 42 47 57 63 68 110 157 198 207 223 260 263 265 320 327 333 347 351
[20] 361 374 381 382 395 397 398 422 447 464 468 522 576 627 631 677 684 732 741
[39] 773 780 781 810 817 820 825 834 862 871 879 887 888 922 940 946 957 973 989

> which(myresult_F$permutation_p[, 3]<=0.05)

[1]  6 42 47 53 57 62 63 82 113 157 184 207 219 223 258 260 263 265 271
[20] 327 333 347 351 361 374 381 382 395 397 398 422 464 468 517 518 522 576 627
[39] 631 677 732 733 780 781 810 820 825 834 879 886 888 900 922 946 980 997

> con1_adjp <- p.adjust(myresult_F$permutation_p[, 1], "BH")
> sum(con1_adjp<=0.05/3)

[1] 4

```

```

> con2_adjp <- p.adjust(myresult_F$permutation_p[, 2], "BH")
> sum(con2_adjp<=0.05/3)

[1] 17

> con3_adjp <- p.adjust(myresult_F$permutation_p[, 3], "BH")
> sum(con3_adjp<=0.05/3)

[1] 5

> which(con2_adjp<=0.05/3)

[1] 57 63 207 260 265 351 374 381 382 468 576 627 677 810 879 888 946

> which(con3_adjp<=0.05/3)

[1] 63 398 468 677 888

> unifdata_F <- matrix(runif(1000*18, 0.15, 0.98), 1000, 18)
> mydesign2_F <- c(rep(0, 6), rep(1, 6), rep(2, 6))
> aContrast <- c("X1-X0", "X2-X1", "X2-X0")
> myresult2_F <- RBM_F(unifdata_F, mydesign2_F, aContrast, 100, 0.05)
> summary(myresult2_F)

              Length Class  Mode
ordfit_t      3000   -none- numeric
ordfit_pvalue 3000   -none- numeric
ordfit_beta1  3000   -none- numeric
permutation_p 3000   -none- numeric
bootstrap_p   3000   -none- numeric

> sum(myresult2_F$bootstrap_p[, 1]<=0.05)

[1] 63

> sum(myresult2_F$bootstrap_p[, 2]<=0.05)

[1] 56

> sum(myresult2_F$bootstrap_p[, 3]<=0.05)

[1] 56

> which(myresult2_F$bootstrap_p[, 1]<=0.05)

[1] 18 20 21 35 40 69 77 83 89 91 93 131 179 215 218 222 224 227 229
[20] 238 239 281 297 307 344 404 410 457 481 494 497 500 532 547 551 555 565 579
[39] 607 630 634 639 665 672 688 690 692 701 706 713 718 738 743 757 818 831 853
[58] 869 922 923 940 954 957

```

```

> which(myresult2_F$bootstrap_p[, 2]<=0.05)

[1] 18 20 21 30 35 40 69 77 83 89 91 93 98 179 187 218 224 227 334
[20] 344 368 402 404 410 457 481 494 497 500 532 547 555 565 579 607 630 634 639
[39] 672 688 690 692 701 706 713 718 740 743 757 818 831 853 922 923 957 959

> which(myresult2_F$bootstrap_p[, 3]<=0.05)

[1] 18 20 21 30 40 69 77 83 89 91 93 131 179 218 222 224 227 238 306
[20] 344 402 404 410 430 457 481 494 500 532 538 547 551 555 565 579 607 630 634
[39] 639 672 688 690 701 740 757 818 831 834 853 869 922 923 940 957 959 993

> con21_adjp <- p.adjust(myresult2_F$bootstrap_p[, 1], "BH")
> sum(con21_adjp<=0.05/3)

[1] 13

> con22_adjp <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adjp<=0.05/3)

[1] 8

> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=0.05/3)

[1] 13

```

4 Ovarian cancer methylation example using the RBM_T function

Two-group comparisons are the most common contrast in biological and biomedical field. The ovarian cancer methylation example is used to illustrate the application of RBM_T in identifying differentially methylated loci. The ovarian cancer methylation example is taken from the genome-wide DNA methylation profiling of United Kingdom Ovarian Cancer Population Study (UKOPS). This study used Illumina Infinium 27k Human DNA methylation Beadchip v1.2 to obtain DNA methylation profiles on over 27,000 CpGs in whole blood cells from 266 ovarian cancer women and 274 age-matched healthy controls. The data are downloaded from the NCBI GEO website with access number GSE19711. For illustration purpose, we chose the first 1000 loci in 8 randomly selected women with 4 ovarian cancer cases (pre-treatment) and 4 healthy controls. The following codes show the process of generating significant differential DNA methylation loci using the RBM_T function and presenting the results for further validation and investigations.

```

> system.file("data", package = "RBM")

[1] "/tmp/RtmpubtCbR/Rinst35701b1f023b15/RBM/data"

> data(ovarian_cancer_methylation)
> summary(ovarian_cancer_methylation)

```

IlmnID	Beta	exmdata2[, 2]	exmdata3[, 2]
cg00000292: 1	Min. :0.01058	Min. :0.01187	Min. :0.009103
cg00002426: 1	1st Qu.:0.04111	1st Qu.:0.04407	1st Qu.:0.041543
cg00003994: 1	Median :0.08284	Median :0.09531	Median :0.087042
cg00005847: 1	Mean :0.27397	Mean :0.28872	Mean :0.283729
cg00006414: 1	3rd Qu.:0.52135	3rd Qu.:0.59031	3rd Qu.:0.558575
cg00007981: 1	Max. :0.97069	Max. :0.96937	Max. :0.970155
(Other) :994		NAs :4	

exmdata4[, 2]	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]
Min. :0.01019	Min. :0.01108	Min. :0.01937	Min. :0.01278
1st Qu.:0.04092	1st Qu.:0.04059	1st Qu.:0.05060	1st Qu.:0.04260
Median :0.09042	Median :0.08527	Median :0.09502	Median :0.09362
Mean :0.28508	Mean :0.28482	Mean :0.27348	Mean :0.27563
3rd Qu.:0.57502	3rd Qu.:0.57300	3rd Qu.:0.52099	3rd Qu.:0.52240
Max. :0.96658	Max. :0.97516	Max. :0.96681	Max. :0.95974
	NAs :1		

exmdata8[, 2]
Min. :0.01357
1st Qu.:0.04387
Median :0.09282
Mean :0.28679
3rd Qu.:0.57217
Max. :0.96268

```

> ovarian_cancer_data <- ovarian_cancer_methylation[, -1]
> label <- c(1, 1, 0, 0, 1, 1, 0, 0)
> diff_results <- RBM_T(aData=ovarian_cancer_data, vec_trt=label, repetition=100, alpha=0.05)
> summary(diff_results)

```

	Length	Class	Mode
ordfit_t	1000	-none-	numeric
ordfit_pvalue	1000	-none-	numeric
ordfit_beta0	1000	-none-	numeric
ordfit_beta1	1000	-none-	numeric
permutation_p	1000	-none-	numeric
bootstrap_p	1000	-none-	numeric

```
> sum(diff_results$ordfit_pvalue<=0.05)
```

```
[1] 47
```

```
> sum(diff_results$permutation_p<=0.05)
```

```
[1] 33
```

```
> sum(diff_results$bootstrap_p<=0.05)
```

```
[1] 57
```

```
> ordfit_adjp <- p.adjust(diff_results$ordfit_pvalue, "BH")  
> sum(ordfit_adjp<=0.05)
```

```
[1] 0
```

```
> perm_adjp <- p.adjust(diff_results$permutation_p, "BH")  
> sum(perm_adjp<=0.05)
```

```
[1] 3
```

```
> boot_adjp <- p.adjust(diff_results$bootstrap_p, "BH")  
> sum(boot_adjp<=0.05)
```

```
[1] 0
```

```
> diff_list_perm <- which(perm_adjp<=0.05)  
> diff_list_boot <- which(boot_adjp<=0.05)  
> sig_results_perm <- cbind(ovarian_cancer_methylation[diff_list_perm, ], diff_results$ordfit_t[diff_list_perm, ])  
> print(sig_results_perm)
```

	IlmnID	Beta	exmdata2[, 2]	exmdata3[, 2]	exmdata4[, 2]
19	cg00016968	0.80628480	NA	0.81440820	0.83623180
627	cg00612467	0.04777553	0.03783457	0.05380982	0.05582291
764	cg00730260	0.90471270	0.90542290	0.91002680	0.91258610
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
19	0.80831380	0.73306440	0.82968340	0.8491780	
627	0.04740551	0.05332965	0.05775211	0.0557971	
764	0.90575890	0.88760470	0.90756300	0.9094679	
	diff_results\$ordfit_t[diff_list_perm]				
19	-2.547097				
627	-1.797392				
764	-1.560713				
	diff_results\$permutation_p[diff_list_perm]				
19	0				
627	0				
764	0				

```
> sig_results_boot <- cbind(ovarian_cancer_methylation[diff_list_boot, ], diff_results$ordfit_t[diff_list_boot, ])  
> print(sig_results_boot)
```

```
[1] IlmnID  
[2] Beta  
[3] exmdata2[, 2]  
[4] exmdata3[, 2]  
[5] exmdata4[, 2]
```

```
[6] exmdata5[, 2]
[7] exmdata6[, 2]
[8] exmdata7[, 2]
[9] exmdata8[, 2]
[10] diff_results$ordfit_t[diff_list_boot]
[11] diff_results$bootstrap_p[diff_list_boot]
<0 rows> (or 0-length row.names)
```